

АВТОМАТИЗИРОВАННАЯ СИСТЕМА РАСПРЕДЕЛЁННЫХ ВЫЧИСЛЕНИЙ ДЛЯ ЧИСЛЕННОГО РЕШЕНИЯ АЛГЕБРАИЧЕСКИХ ЗАДАЧ

Е.А. Корчевская

доцент, к.ф.-м.н., e-mail: korchevskaya.elena@tut.by

Л.В. Маркова

доцент, к.ф.-м.н., e-mail: l_v_markova@mail.ru

А.Н. Красоткина

преподаватель кафедры прикладной математики и механики,
e-mail: krasotkina_anna030288@mail.ru

Витебский государственный университет имени П.М. Машерова

Аннотация. Авторами предлагается новый подход к решению основных задач вычислительных методов алгебры на основании принципов объектно-ориентированного проектирования и способов создания параллельных алгоритмов. Все вычислительно-конструктивные понятия линейной алгебры рассматриваются в качестве объектов и представлены единой классификационной иерархией.

Ключевые слова: алгоритмы, параллелизм, программирование, потоки.

Широкое внедрение идей параллелизма в работу вычислительных систем является наиболее перспективным и динамичным направлением увеличения скорости решения прикладных задач.

В настоящее время существует большое количество разнообразных технологий организации параллелизма при разработке приложений в системах с общей памятью. Все они различаются моделями исполнения и планирования распределения нагрузки. От специфики реализуемого алгоритма в большинстве случаев зависит целесообразность применения конкретной технологии.

Создание потоков — это базовый способ достижения параллелизма в приложениях, разрабатываемых для систем с общей памятью. Параллелизм на уровне ядер процессора обеспечивается многопоточностью. Потоки могут решать как однотипные задачи на разных наборах входных данных, так и разнотипные.

Решение больших систем линейных уравнений принадлежит к кругу задач, для которых применение новейшей высокопроизводительной вычислительной техники дает максимальную эффективность.

Придерживаясь основных принципов методологии объектно-ориентированного программирования, авторы разработали иерархии матричных классов и алгоритмов линейной алгебры [1]. Создание конкретного матричного класса сводится к реализации набора операций с элементарными матрицами. Значительная часть операций непосредственно реализуется в верхних

матричных классах. Тем не менее, переопределение их в конкретных классах с учётом частных математических свойств, особенностей конкретных структур данных позволяет при необходимости добиться наибольшей эффективности программного кода. Предложенная авторами классификация элементарных матриц определяет базовый набор матричных преобразований, участвующих в методах линейной алгебры [1].

Матричные операции являются базовыми макрооперациями для многих задач линейной алгебры. Поэтому эффективный параллельный алгоритм операций над матрицами позволяет значительно увеличить размерность подобных задач без увеличения времени их решения.

Используя сформированные иерархические структуры, авторы описали и реализовали алгоритмы распараллеливания основных матричных операций на языке программирования C++ с использованием технологии объектно-ориентированного программирования.

В качестве примера приведём схему параллельного алгоритма произведения квадратных матриц, основанного на блочном представлении матриц:

1. Распределяем по процессорам блоки матриц A , B так, как показано на рисунке 1.
2. Параллельно на всех процессорах системы производим вычисление компонент соответствующих блоков матрицы C .
3. Передаём на один из процессоров все вычисленные блоки и формируем матрицу C .
4. Заметим, кроме того, что легко предложить множество модификаций данного алгоритма:
 - вместо строкового распределения блоков матрицы A по процессорам можно использовать распределение по столбцам;
 - можно использовать разбиение матриц A , B на блоки таким образом, чтобы общее количество блоков было равно количеству процессоров в системе, и каждому процессору назначать вычисление одного блока.

Существует два подхода к анализу времени работы алгоритма: теоретический и практический. В основе теоретического подхода лежит построение и доказательство аналитических оценок трудоёмкости. Данный подход теоретически обоснован, абстрагирован от конкретной вычислительной архитектуры и деталей программной реализации, однако, при таком подходе получить достаточно точные оценки времени работы алгоритма в секундах не возможно. В некоторых случаях необходимо оценить время в стандартных единицах — секундах, минутах, часах, опираясь на конкретную вычислительную архитектуру и средства программирования. В таких ситуациях приходится прибегать к вычислительным экспериментам.

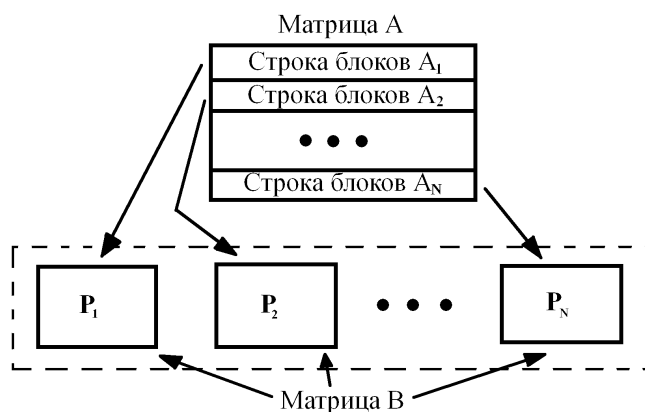


Рис. 1. Блок-схема алгоритма параллельного произведения квадратных матриц

При выполнении вычислительного эксперимента необходимо учитывать следующие моменты: грамотно реализовать замеры времени, подобрать репрезентативные наборы тестовых данных, на которых будут производиться измерения, программную реализацию выбранного алгоритма необходимо адаптировать для вычислительной архитектуры и т.д.

Для анализа времени работы приведённого выше алгоритма был выбран практический подход. Выполнялась серия экспериментов с различными параметрами: размерность матрицы, количество задействованных потоков.

Приведём результаты сравнения последовательного и параллельного умножения квадратных матриц:

Таблица 1. Результаты вычислительного эксперимента

Размерность матрицы	10	50	100	200	1000
Время последовательного умножения	0	0	0.016	0.094	23.744
Время параллельного умножения (кол-во потоков)	0.005(2)	0.005(5)	0.016(5)	0.01033(65)	0.1(75)

Из таблицы видно, что параллельное умножение квадратных матриц имеет преимущество при большой размерности матрицы. В скобках указано оптимальное количество потоков, необходимых для наилучшей работы параллельного алгоритма.

ЛИТЕРАТУРА

1. Маркова Л.В., Корчевская Е.А., Красоткина А.Н. Объектная реализация методов вычислительной алгебры // Вестник Витебского государственного университета. 2013. В. 74, № 2. С. 18–22.

AUTOMATED SYSTEM FOR DISTRIBUTED COMPUTING FOR NUMERICAL SOLUTION OF ALGEBRAIC PROBLEMS

E.A. Korchevskaya

Associate Professor, Ph.D. (Math.), e-mail: korchevskaya.elena@tut.by

L.V. Markova

Associate Professor, Ph.D. (Math.), e-mail: l_v_markova@mail.ru

A.N. Krasotkina

Senior Teacher, e-mail: krasotkina_anna030288@mail.ru

Vitebsk State University named after P.M. Masherov

Abstract. The authors propose a new approach to solving the fundamental problems of computational algebra methods based on the principles of object-oriented design and how to create parallel algorithms. All computationally design concepts of linear algebra are treated as objects and represented by a single classification hierarchy.

Keywords: algorithms, concurrency, programming streams.